

デジタルコンテンツ 仕様書

チーム名
フジシロクエスト！

メンバー

豊田 裕也
藤代 健吾
逸見 優聡
別所 一洸
長谷川 文

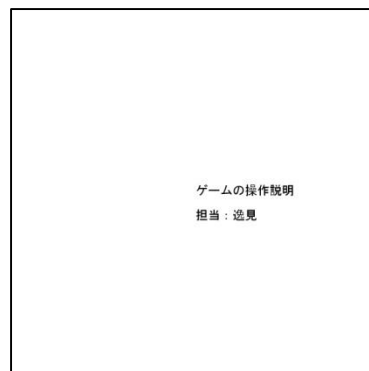
-----外部仕様-----

① タイトル画面



スペースキーで②の画面へ移動。

② ゲームの操作説明



スペースキーで③の画面へ移動。

③ キャラクターセレクト



十字キーで好きなキャラクターを選択。
スペースキーで決定し、④の画面へ移動。

④ プロローグ

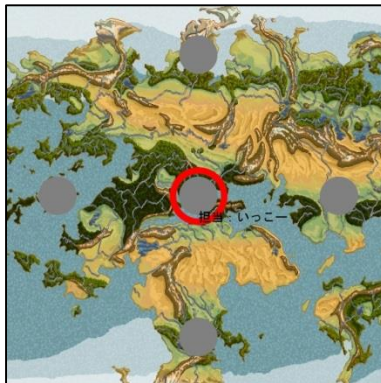
昔々、偉い人がビック・サンを建設した。
ビック・サンは人口の9割が学生で構成されている国である。
2011年4月 フジシロはビック・サンを訪れた。
友達はそこまですくなく、彼女はいなかったが
充実した日々を過ごしていた。
しかし、ある日フジシロは自分の単位数が危ないことに気が付いた
このままでは、留年が決まってしまう。

この物語はフジシロが単位を獲得する為に
様々な試験を乗り越えていくお話である。



スペースキーで⑤の画面へ移動。

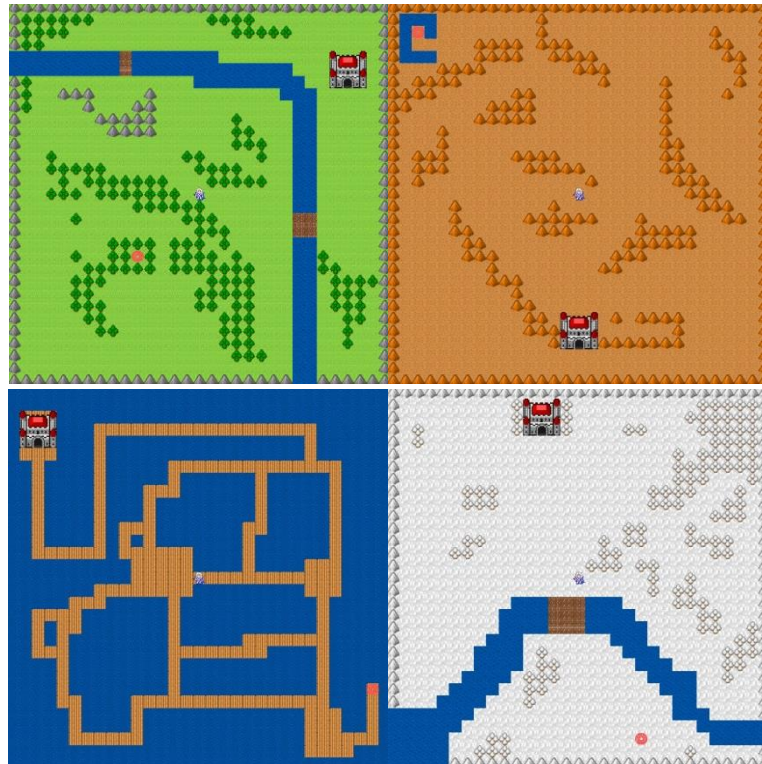
⑤ ワールドマップ



十字キーで上下左右+真ん中の中からダンジョンを選択する。

スペースキーで決定し、⑥の画面へ移動

⑥ ダンジョンマップ



十字キーでキャラクターの移動。

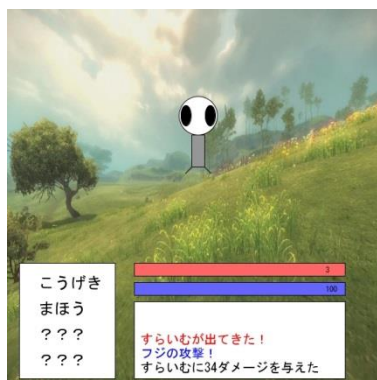
スペースキーでステータス画面の表示。

移動するとランダムで敵が出現し、⑦の画面へ移動。

マップの出口(赤い円陣)まで移動すると、⑤の画面へ移動。

特定オブジェクト(城の入り口)に接触すると、ボス戦となり、⑦の画面へ移動。

⑦ 戦闘画面

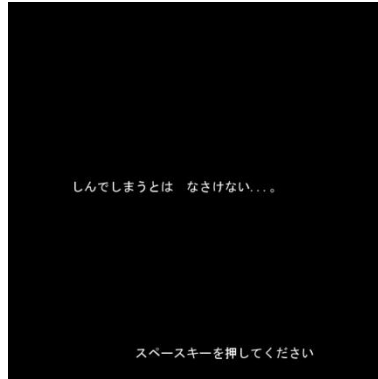


十字キーでアクションを選択、スペースで決定。

戦闘により相手の HP を 0 にするか、「にげる」コマンドを選択することで⑥の画面へ移動。

自分の HP が 0 になると、ゲームオーバーとなり、⑧の画面へ移動。

⑧ ゲームオーバー



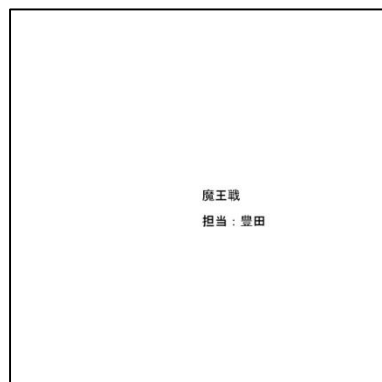
スペースキーで⑤の画面へ移動。

⑨ 魔王戦

デジタルコンテンツの単位を取得するために戦ってきたフジシロ
ウツディ、フミ、ヘンミ、イツコーによる
誘惑や邪魔による攻撃に負けず
デジタルコンテンツの単位を占める魔王テツロウを撃破し
見事フジシロは単位を獲得することに成功。
これで無事にデジタルコンテンツの単位を獲得し
留年は阻止できたフジシロは
残りの学生生活を楽しんだのであった。

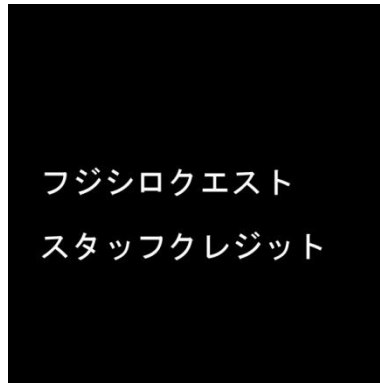


⑩ エピローグ

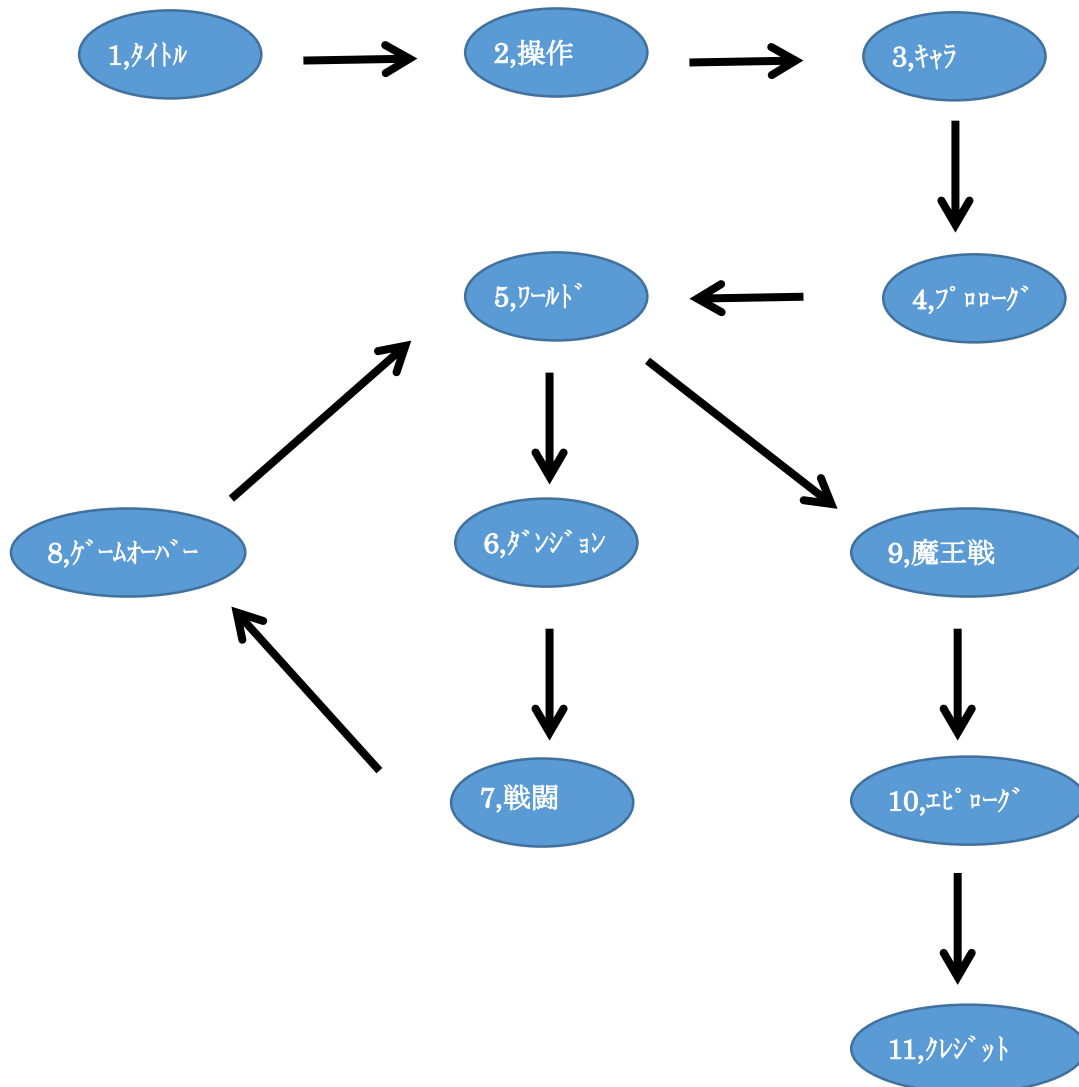


スペースキーで⑩の画面へ移動。

⑪ スタッフクレジット



プログラム終了



-----内部仕様書-----

◎クラス

「作成」にはクラスの作成を担当する責任者名を記載。

「編集」にはそのクラスの内容を一部編集したことのある人の名前を記載。

○Player クラス（作成：藤代健吾 編集：豊田裕也）

プレイヤーが操作するキャラクター（主人公）に関するクラス

・変数

int 型

hp：体力

max_Hp：体力上限値

mp：マジックポイント。魔法は mp を消費することで発動する。

max_Mp：mp の上限値

a：攻撃力

b：防御力

exp：取得経験値

level：キャラクターのレベル

type：成長タイプを記録する。

1:体力タイプ

2:魔法タイプ

3:防御タイプ

4:攻撃タイプ

Boolean 型

skill：覚えているスキルを記録する boolean 型配列。

・コンストラクタ

受け取った数字(=キャラクターセレクトで選択されたタイプ)によって

キャラクターの成長タイプの数字を type 変数に記録し、

HP、MP、攻撃力(a)、防御力(b)の各初期能力値を決定する。

・メソッド

Boolean isDead()

⇒プレイヤーの HP が 0 以下になった時、true を返す。

Boolean level_up(int e)

⇒倒した敵の経験値を引数として受け取り、プレイヤーの経験値に可算する。

⇒経験値がある一定値を超えていればレベルを 1 増やし、true を返す。

○Magic クラス （作成：藤代健吾 編集：豊田裕也）

プレイヤーが使用する魔法の情報を入力するクラス

・変数

String 型

name : 名前を入力する String 型変数

int 型

a : 攻撃力を入力する（回復魔法の場合、回復量。能力変化系の場合、変化量）

mp : 消費 MP

type : 種類の分別。

1:攻撃魔法

2:HP 回復

3:MP 回復

4:攻撃力強化

5:防御力強化

・コンストラクタ

String 型の文章を受け取り、name に代入する。

int 型の数字を 3 つ受け取り、それぞれ a、mp、type に代入する。

○Enemy クラス （作成：逸見優聡）

敵の情報を入力するクラス

・変数

int 型

species : 種類を分別するクラス

hp : 体力

max_Hp : 体力上限値

mp : マジックポイント。魔法は mp を消費することで発動する。

max_Mp : mp の上限値

a : 攻撃力

b : 防御力

magic : 魔法攻撃力

recovery : 回復魔法使用時の回復量

exp : この敵を倒すことで得られる経験値

String 型

name : キャラクターの名前

attack_name : 攻撃技の名前

・コンストラクタ

数字の一つ受け取り、その数字によって敵の種類を分別する。

0 : 弱い敵

1 : HP の高い敵

2 : 攻撃力の高い敵

3 :

4 :

5 :

100 :

200 :

300 :

400 :

500 :

・メソッド

`int attack_type()`

⇒敵の攻撃時に発動。

⇒条件によって攻撃技名(`attack_name`)を決定し、その種類によって数字を返す。

10 : 通常攻撃

11 : 魔法攻撃

12 : 回復魔法

`Boolean isDead()`

⇒敵の HP が 0 以下になった時、`true` を返す。

`void display(int x, int y)`

⇒戦闘中に敵が生きている間、指定された座標に敵のグラフィックを表示。

⇒敵の種類(`species`)によって、どのグラフィックを表示するかを決定する。

○Select クラス (作成 : 豊田裕也)

ワールドマップで、どのエリアを選択しているのかを表示する。

・変数

`int` 型

`position` : どのエリアを指定しているのかを記録。

0 : 中央

1 : 右 (東エリア)

2 : 下 (南エリア)

3 : 左 (西エリア)

4 : 上 (北エリア)

`x` : 選択しているエリアに表示する○の `x` 座標

`y` : 選択しているエリアに表示する○の `y` 座標

・メソッド

`void move(int a)`

⇒キーが入力されたときに発動

⇒入力されたキーによって、引数の値が変わる。

1 : 上

2 : 下

3 : 右

4 : 左

⇒その引数によって **position** の値を以下の状態遷移表に従って変化させる。

Position／引数	1(上)	2(下)	3(右)	4(左)
0(中央)	4(北)	2(南)	1(東)	3(西)
1(東)	4(北)	2(南)	×	0(中央)
2(南)	0(中央)	×	1(東)	3(西)
3(西)	4(北)	2(南)	0(中央)	×
4(北)	×	0(中央)	1(東)	3(西)

`void display()`

⇒ポジションの値によって **x,y** 座標の値を決定する。

⇒決定した座標に○を描く。

○Select_chara クラス（作成：藤代健吾）

キャラクターセレクト時にどのキャラを選択しているかを表示する。

・変数

int 型

position：どのエリアを指定しているのかを記録。

0：左上(攻撃タイプ)

1：右上(防御タイプ)

2：左下(魔法タイプ)

3：右下(体力タイプ)

x：選択しているエリアに表示する四角形の x 座標

y：選択しているエリアに表示する四角形の y 座標

・メソッド

void move(int a)

⇒キーが入力されたときに発動。

⇒入力されたキーによって、引数の値が変わる。

1：上

2：下

3：右

4：左

⇒その引数によって position の値を以下の状態遷移表に従って変化させる。

Position／引数	1(上)	2(下)	3(右)	4(左)
0(左上)	×	2(左下)	1(右上)	×
1(右上)	×	3(右下)	×	0(左上)
2(左下)	0(左上)	×	3(右下)	×
3(右下)	1(右上)	×	×	2(左下)

void display()

⇒ポジションの値によって x,y 座標の値を決定する。

⇒決定した座標に四角形を描く。

○MAPSb クラス (作成：別所一洗)

2次元配列でダンジョンマップを書く。

{0,1,0}

{1,1,1}

{0,2,0}

0 = 山 1 = 平地 2 = 川 と変換

{山,平,山}

{平,平,平}

{山,川,山}

・変数

Float 型変数

SZ : 画面サイズを受け取る

L : Map の 2次元配列の長さを受け取る

def : マップを描く時の 1 マスの辺長 (SZ/L)

selx : イベントの選択肢の位置を決め。YES / NO で位置が変わるので

・メソッド

void drawMap(int x,int y,int z)

⇒MAP とプレイヤーを描画するクラス。

⇒変数 z でステージを判別する。

⇒2次元配列の行列番号 x,y を引数に、一辺の長さが def の正方形で画像を表示する。

⇒また 2次元配列の中身により描くものを識別している。

⇒その後プレイヤーと boss の城を描画している。

void event(int pt)

イベント番号 pt を受け取りそのイベントの状態にする

0 : 何も起こらない (通常の状態)

1 : 中ボスと戦うかどうか

2 : WorldMap に帰還するかどうか? (帰還ポイント)

○Command1 クラス & Command2 クラス (作成：豊田裕也)

戦闘時、「たたかう」「まほう」「にげる」を選択するときに表示する矢印

・変数

Int 型

position：どの項目を選択しているのかを記録する。

Command1

0：たたかう

1：まほう

2：???

3：にげる

x：矢印を表示する x 座標

y：矢印を表示する y 座標

max：選択肢の数

・コンストラクタ

変数 max に選択肢の数を入力する。

・メソッド

void move(int a)

⇒キー入力によって変数を受け取る。

下を入力：1

上を入力：-1

⇒変数 position に受け取った変数を足し合わせる。

void display()

⇒変数 position の値によって、x,y の座標を変化させる。

⇒x,y の座標に合わせて、矢印を表示する。

◎グローバル変数

○int 型

menu : ゲームの進行状態。

- 0 : タイトル画面
- 1 : ゲームの操作説明
- 2 : キャラセレクト
- 3 : プロローグ
- 4 : ワールドマップ
- 5 : ダンジョンマップ
- 6 : 魔王戦
- 7 : エピローグ
- 8 : スタッフレジット
- 9 : 戦闘

SZ : マップ表示用に画面サイズを代入する。

mapx : プレイヤーの X 座標

mapy : プレイヤーの Y 座標

maptime : MAP の時間 1/60 秒で+1

map_movetime:プレイヤーの移動時間 1/60 秒で+1 プレイヤーの移動速度を決める

Event : 発生するイベントの番号

map_manutime : ステータス表示の ON/OFF の切り替える時間 の目安

MiniMap[][] : MAP の設計図 MapSub クラス drawMap で使用

ed_y : スタッフレジットを表示する y 座標。

ed_time : クレジットをスクロールするタイミングを計る。

battle : 戦闘の進行。

100 : 戦闘開始のログを流す。

101 : 誤操作防止のため、決定キー入力されるまで待機

0 : 主人公の行動を決定する。

1 : 使う魔法を決定する。

2 : 主人公が決定した行動の計算処理とログ更新

3 : 敵の死亡判定

4 : 敵の行動の計算処理とログ更新

5 : 主人公の死亡判定

6 : 戦闘終了後のログ表示と取得経験値の処理

7 : 決定キーが押されるまで待機。戦闘用変数を初期化し、ダンジョンマップに戻る。

999 : 戦闘敗北

damage : 主人公と敵の攻撃によって発生するダメージ数値。

critical : 攻撃のクリティカル発生を判定するため、ランダムで数値を代入。

gameover_message : 戦闘敗北時に表示する文章を判定するため、ランダムで数値を代入。

player_a_rank, player_b_rank : 主人公の能力補正ランク

enemy_a_rank, enemy_b_rank : 敵の能力補正ランク

player_battle_a, player_battle_b : 主人公のダメージ計算時の能力値

enemy_battle_a, enemy_battle_b : 敵のダメージ計算時の能力値

○float 型

encounter : 敵の出現を判定するため、ランダムで数値を代入。

○String 型

log[] : 戦闘時の行動を文章化して代入する配列。

○color 型

font_color : プロログ画面で表示する文字の色。

log_color[] : 戦闘時に表示するコマンドログの文字色。

○PImage 型

kougeki, bougyo, maho, tairyoku : キャラクタセレクト時のキャラクタグラフィック。

field : 戦闘時の背景。

img1, img2, img3 : プロローグ、エピローグで使用する背景。

worldmap : ワールドマップ選択時に使用する背景。

ground1,ground2,ground3,ground4,

river,mountain1,mountain2,mountain4,br,

mori1,mori4,mplayer,sea,mapesc,boss,cas : ダンジョンマップ時に使用する背景。

○boolean 型

up, down, right, left, space : それぞれ対象のキーの入力を判定する。

MapClear[] : クリアしたダンジョンを記録する。

eve1,eve2 : それぞれのイベントの ON OFF

mapmenu : ステータス表示の ON OFF の

fogMessage : 霧に覆われ MAP に進めない時のメッセージの ON OFF

◎関数

「作成」には作成を担当する責任者名を記載。

「編集」には内容を一部編集したことのある人の名前を記載。

○void setup()

⇒画面サイズの決定。

⇒使用するフォント(MS Gothic)の設定。

⇒PImage 型変数に画像ファイルを取り込んで代入。

⇒その他各変数の初期化。

○void draw()

⇒変数 menu の値によって実行する関数を選択し、画面表示を切り替える。

○void keyPressed()

⇒キーを押した時の動作をまとめる。

○void keyReleased()

⇒キーを話した時の動作をまとめる。

○void skill_ac() (作成：藤代健吾 編集：豊田裕也)

⇒主人公が使用する魔法の初期設定を行う。

○void title() (作成：長谷川文 編集：逸見優聡)

⇒タイトル画面を表示する。

⇒決定キーで次の画面に移動する。

○void readme() (作成：逸見優聡)

- ⇒ゲームの操作説明画面を表示する。
- ⇒使用するキーの説明と画面表示の説明。
- ⇒決定キーで次の画面に移動する。

○void charaSelect() (作成：藤代健吾)

- ⇒主人公のキャラクターセレクト画面を表示する。
- ⇒主人公のタイプは攻撃、防御、魔法、体力の4種類。
- ⇒画像と矢印キーで移動する選択用カーソルを表示する。
- ⇒決定キーで次の画面に移動する。

○void prologue() (作成：長谷川文 編集：逸見優聡)

- ⇒ストーリーのプロローグを表示する。
- ⇒決定キーで次の画面に移動する。

○void world() (作成：別所一洗 編集：豊田裕也)

- ⇒ワールドマップを表示する。
- ⇒背景に専用画像を表示する。
- ⇒矢印キーで移動する選択用カーソルを表示する。
- ⇒決定キーで選択したダンジョンに移動する。

○void minimap() (作成：別所一洗)

- ⇒ダンジョンマップを表示する。
- ⇒決定キーでステータスを表示する。

○void mapmove(int movepoint,int direction) (作成：別所一洗)

- ⇒ダンジョンマップで主人公の位置座標を移動する。
- ⇒実行される度に、敵の出現判定を行う。

○void mapmenu_d() (作成：別所一洗)

⇒主人公のステータス画面を表示する。

○void encount_setup() (作成：豊田裕也 編集：逸見優聡)

⇒出現する敵の種類をランダムで決定する。

⇒出現率の設定を書き込んでおく。

⇒決定後、戦闘画面に移動する。

○void boss() (作成：豊田裕也 編集：逸見優聡)

⇒ボスの初期設定を行い、戦闘画面に移動する。

○void epilogue() (作成：長谷川文 編集：逸見優聡)

⇒ストーリーのエピローグを表示する。

○void ending() (作成：豊田裕也)

⇒制作に関わったスタッフのクレジットを表示する。

⇒素材元を表示する。

○void fight() (作成：豊田裕也)

⇒戦闘画面用背景を表示する。

⇒ステータスとバトルログを表示するための関数を実行する。

⇒敵のキャラクターグラフィックを表示する。

⇒変数 battle の値によって画面表示を切り替える。

○void status1() (作成：豊田裕也 編集：藤代健吾)

⇒主人公の行動を決定するスペース枠を表示する。

⇒行動内容(たたかう、まほう、???、にげる)を表示する。

⇒まほうを選択された場合、追加で使用する魔法を決定するスペース枠を表示する。

○void status2()（作成：豊田裕也）

- ⇒HP と MP を表示する用に空のバーを 2 本表示する。
- ⇒それぞれの最大値に対して、現在値の割合を計算する。
- ⇒HP は赤、MP は青でそれぞれのバーの上から現在値の割合だけ塗りつぶし表示する。
- ⇒それぞれの現在値を数値で表示する。

○void status3()（作成：豊田裕也）

- ⇒バトルログを表示するための枠を作成する。
- ⇒指定された文字色でバトルログを表示する。

○void log_off()（作成：豊田裕也）

- ⇒バトルログの文章を代入している配列の内容を一つずつずらす。
- ⇒次に代入される文章に対応している文字色を入れる配列変数を初期化する。

○void judg(int a)（作成：豊田裕也 編集：逸見優聡）

- ⇒攻撃方法を判定する。
- ⇒攻撃方法によって、主人公と敵の能力値を用いて変数 **damage** に値を代入する。
- ⇒**damage** の値によって主人公か敵の能力値を変化させる。
- ⇒変化した内容をログ表示させる。

○void gameover(int ccccc)（作成：豊田裕也）

- ⇒引数に対応した文章を画面に表示する。
- ⇒決定キーを押すとワールドマップに戻れるようにする。